

JAVASCRIPT

# PROJECTE 1

MP07-DESENVOLUPAMENT  
D'INTERFICIES

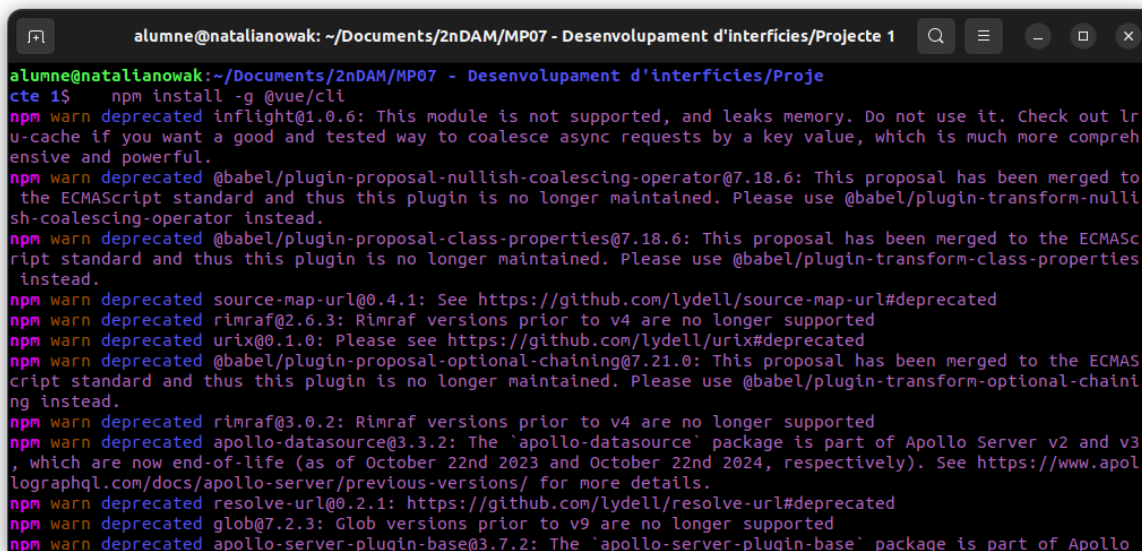
Natalia Nowak

# Índex

<b>Índex.....</b>	<b>1</b>
<b>Instal·lació i configuració.....</b>	<b>2</b>
<b>Pàgines.....</b>	<b>4</b>
Inici.....	4
Calculadora.....	5
Comentaris.....	6
Usuaris.....	8
<b>Testos.....</b>	<b>10</b>
Calculadora.....	10
Comments.....	12
Users.....	13

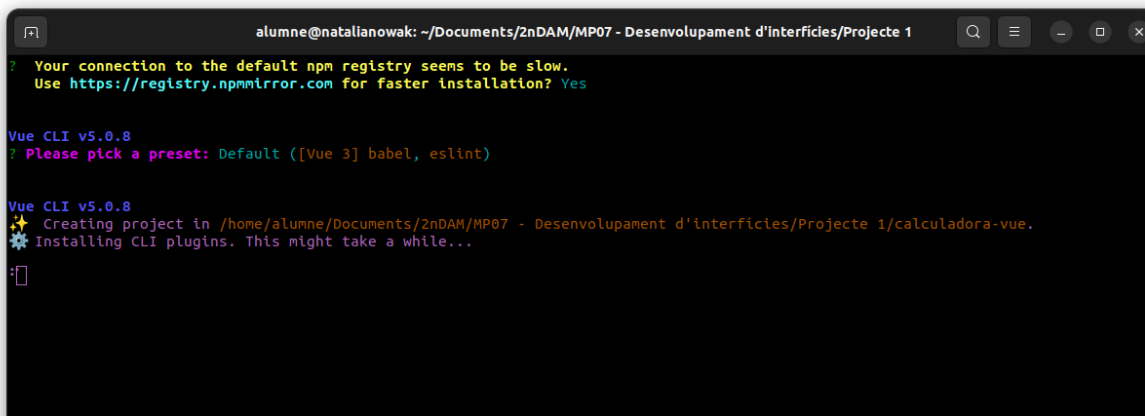
## Instal·lació i configuració

- En primer lloc, tenim que instal·lar `@vue/cli` per a posar-lo en funcionament. Les següents comandes, faig una prova d'instal·lació:
  - Instal·lació VUE en **npm install**:



```
alumne@natalianowak: ~/Documents/2nDAM/MP07 - Desenvolupament d'interfícies/Projecte 1
alumne@natalianowak:~/Documents/2nDAM/MP07 - Desenvolupament d'interfícies/Proje
cte 1$ npm install -g @vue/cli
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-nullish-coalescing-operator instead.
npm warn deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-class-properties instead.
npm warn deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm warn deprecated rimraf@2.6.3: Rimraf versions prior to v4 are no longer supported
npm warn deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm warn deprecated @babel/plugin-proposal-optional-chaining@7.21.0: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-optional-chaining instead.
npm warn deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm warn deprecated apollo-datasource@3.3.2: The 'apollo-datasource' package is part of Apollo Server v2 and v3, which are now end-of-life (as of October 22nd 2023 and October 22nd 2024, respectively). See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm warn deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated apollo-server-plugin-base@3.7.2: The 'apollo-server-plugin-base' package is part of Apollo
```

- Creació projecte calculadora en **create vue calculadora**:



```
alumne@natalianowak: ~/Documents/2nDAM/MP07 - Desenvolupament d'interfícies/Projecte 1
? Your connection to the default npm registry seems to be slow.
  Use https://registry.npmmirror.com for faster installation? Yes

Vue CLI v5.0.8
? Please pick a preset: Default ([Vue 3] babel, eslint)

Vue CLI v5.0.8
✨ Creating project in /home/alumne/Documents/2nDAM/MP07 - Desenvolupament d'interfícies/Projecte 1/calculadora-vue.
⚙ Installing CLI plugins. This might take a while...

📁
```

Després el que cal fer es entrar dins del directori de calculadora (`cd calculadora/`) i executar la comanda **npm run dev**

# Pàgines

## Inici

En la pàgina d'inici podem trobar els següents elements:

### Títol Principal i Descripció:

## Benvingut a la meua SPA de Vue.js

Una aplicació web que mostra tres exemples de Vue.js

Un títol destacat que diu "Benvingut a la meua SPA de Vue.js", que indica l'objectiu de l'aplicació. En la descripció, podem trobar un paràgraf que proporciona una breu descripció de l'aplicació, mencionant que mostra tres exemples de Vue.js.

### Seccions de Contingut:

Tres seccions, cadascuna representant una funcionalitat diferent de l'aplicació:

#### Calculadora

Calculadora amb suport per a diverses operacions

Calculadora

#### Comentaris

Comentaris obtinguts de l'API JSONPlaceholder

Comentaris

#### Usuaris

Dades d'usuaris de l'API DummyJSON

Usuaris

**Calculadora:** Una secció que descriu una calculadora amb suport per a diverses operacions, amb un enllaç que porta a la ruta </calculator>.

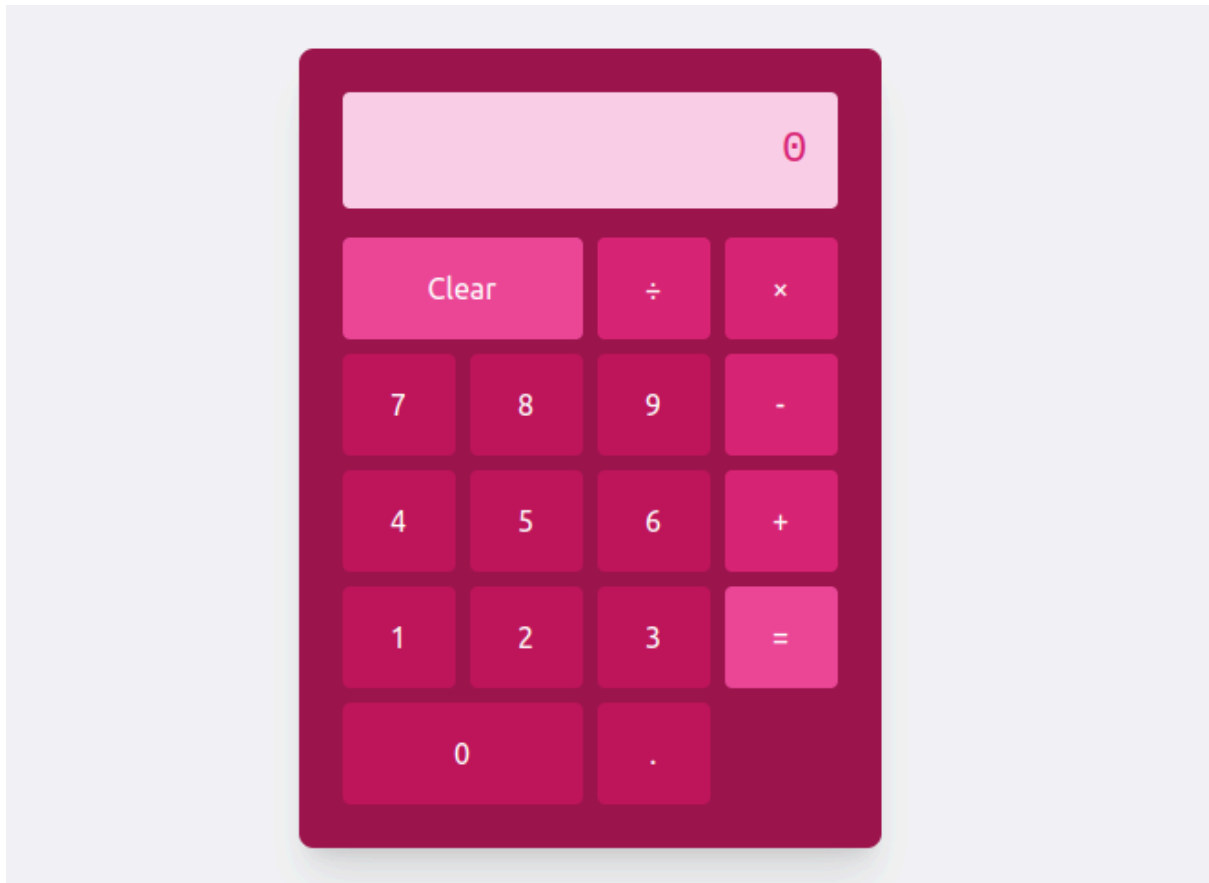
**Comentaris:** Una secció que parla sobre la funcionalitat de mostrar comentaris obtinguts d'una API (JSONPlaceholder), amb un enllaç a la ruta </comments>.

**Usuaris:** Una secció que presenta dades d'usuaris obtingudes d'una altra API (DummyJSON), amb un enllaç a la ruta </users>.

# Calculadora

## Interfície d'Usuari:

Una interfície dissenyada per a facilitar l'ús de la calculadora, amb una disposició clara i accessible.



En aquesta calculadora podem fer totes les operacions bàsiques necessàries. També podem afegir decimals i netejar el contingut en cas de necessitar corregir errors.

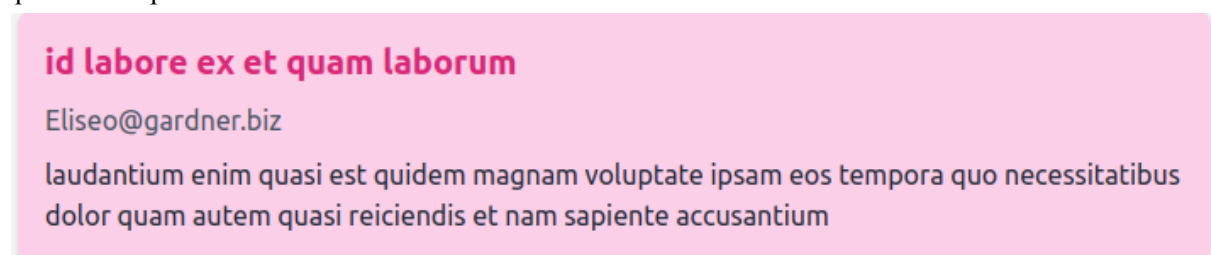
## Comentaris

La secció de Comentaris d'aquest projecte Vue.js té com a objectiu principal mostrar una llista de comentaris obtinguts d'una API externa (en aquest cas, JSONPlaceholder).



### Obtenir Comentaris:

Quan la secció es carrega, s'executa una sol·licitud HTTP a l'API per obtenir una llista de comentaris. Això es fa mitjançant la funció `fetch()` dins del mètode `onMounted`, que s'executa automàticament quan el component es monta.



La **interfície d'usuari** mostra diferents continguts basats en l'estat de càrrega:

- Un missatge de càrrega ("Carregant comentaris...") mentre els comentaris s'estan carregant.

## **Comentaris**

Carregant comentaris...

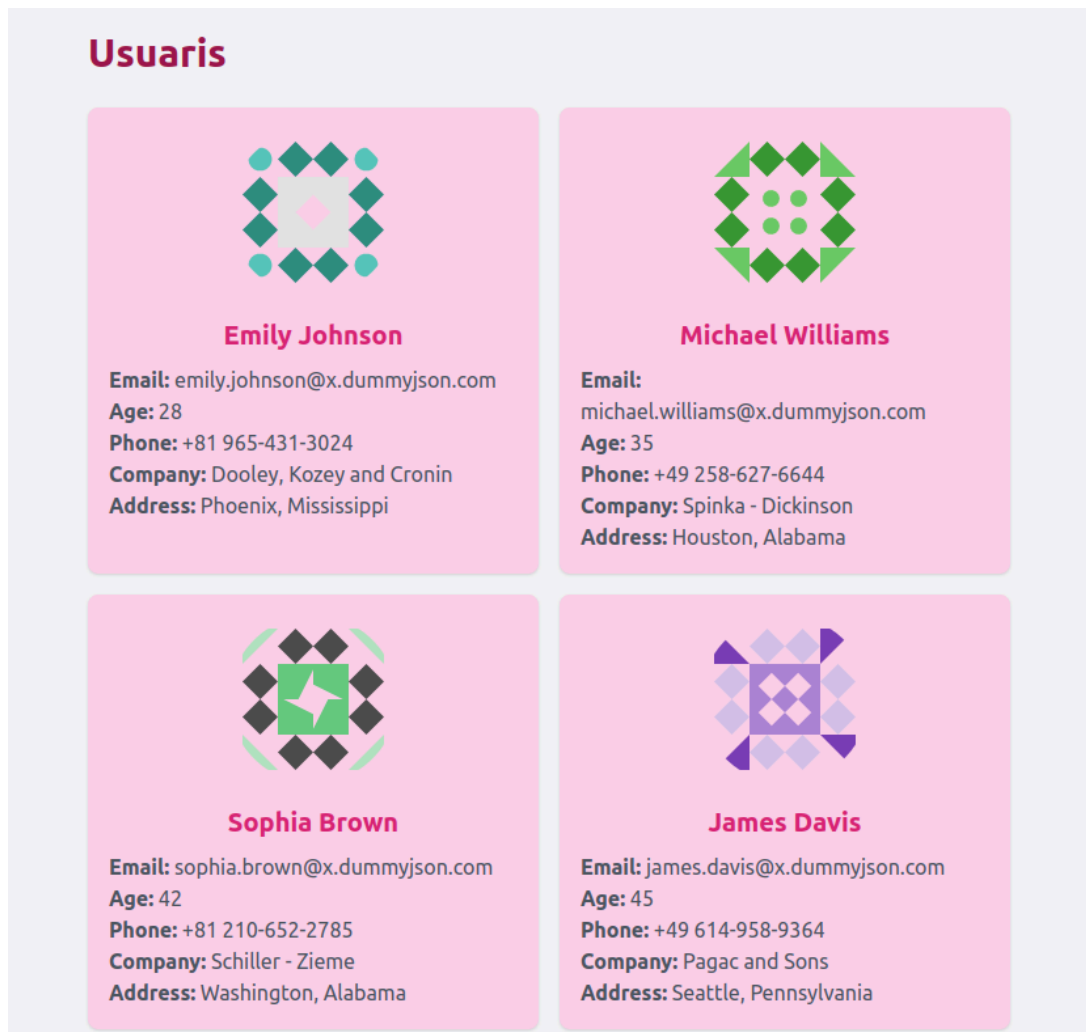
- Un missatge d'error si hi ha un problema en obtenir els comentaris.

## **Comentaris**

Error carregant comentaris

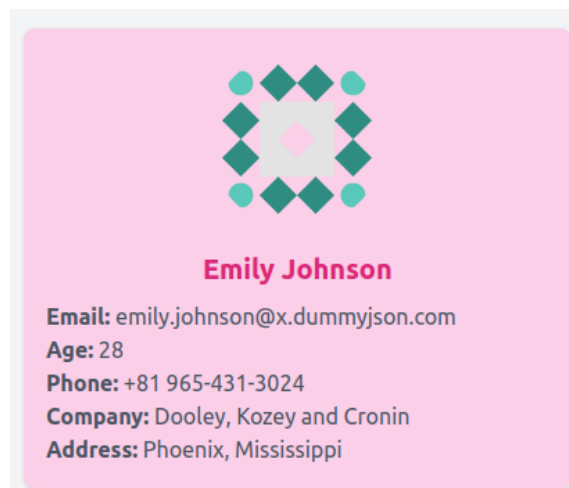
## Usuaris

La secció de Usuaris d'aquest projecte Vue.js té com a objectiu principal mostrar una llista d'usuaris obtinguts d'una API externa (en aquest cas, DummyJSON).



### Obtenir Usuaris:

Quan la secció es carrega, s'executa una sol·licitud HTTP a l'API per obtenir una llista d'usuaris. Això es fa mitjançant la funció `axios.get()` dins del mètode `onMounted`, que s'executa automàticament quan el component es monta.





La **interfície d'usuari** mostra diferents continguts basats en l'estat de càrrega:

## Usuaris

Carregant usuaris...

- Un missatge de càrrega ("Carregant usuaris...") mentre els usuaris s'estan carregant.

## Usuaris

Error carregant usuaris

- Un missatge d'error si hi ha un problema en obtenir els usuaris.

URL

- La url que utilitzarem serà <http://localhost:5173/>



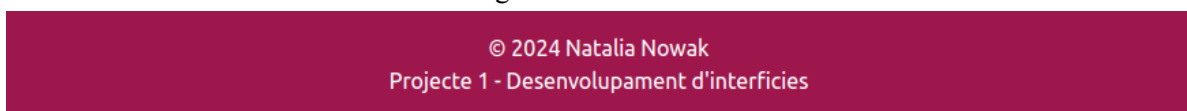
Header

- Conecta en totes les pàgines de l'app



Footer

- Informació sobre l'alumna i l'assignatura



# Testos

## Calculadora

En primer lloc, cal executar la següent comanda per a poder fer test en vue:

```
npm install --save-dev jest @vue/test-utils
```

```
alumine@natalianowak:~/Documents/2nDAM/MP07 - Desenvolupament d'interfícies/Projecte 1/ProjecteJavaScript1/project$ npm install --save-dev jest @vue/test-utils
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
```

Configurem l'arxiu vitest.config.js:

```
import { defineConfig } from 'vitest/config'
import vue from '@vitejs/plugin-vue'

export default defineConfig({
  plugins: [vue()],
  test: {
    environment: 'jsdom',
    globals: true,
  },
})
```

- Executem el test en la següent comanda:

```
npx vitest src/views/test/calculator.spec.js
```

```
RERUN src/views/test/calculator.spec.js x17
```

```
✓ src/views/test/calculator.spec.js (5)  
✓ Calculator.vue (5)  
  ✓ should display the initial value  
  ✓ should add digits correctly  
  ✓ should perform addition  
  ✓ should perform subtraction  
  ✓ should clear the display
```

```
Test Files 1 passed (1)  
Tests      5 passed (5)
```

Com veiem, el test s'ha completat en èxit.

Continguts del test:

- L'arxiu de prova s'organitza en blocs **describe i it**, que són funcions proporcionades per Jest per agrupar i definir proves.
- Cada prova es defineix amb la funció **it**, que pren una descripció de la prova i una funció que conté la lògica de la prova. Algunes de les proves que realitza són les següents:
  - Verificar el valor inicial
  - Afegir dígits correctament
  - Realitzar operacions matemàtiques
  - Netejar la pantalla

Exemples de codi:

- Netejar pantalla:

```
it('should clear the display', async () => {  
  const buttons = wrapper.findAll('button')  
  
  const button7 = buttons.find(button => button.text() === '7')  
  await button7.trigger('click')  
  
  const buttonClear = buttons.find(button => button.text() === 'Clear')  
  await buttonClear.trigger('click')  
  
  expect(wrapper.find('.text-right').text()).toBe('0')  
})
```

## Comments

Per a Comments.vue hem creat un nou arxiu a l'igual que en calculator per a fer els testos necessaris per al seu funcionament, en aquest cas, executem **npx vitest comments.spec.js** i ens surt el següent missatge:

```
RERUN src/views/test/comments.spec.js x1
✓ src/views/test/comments.spec.js (3) 2107ms
  ✓ Comments.vue (3) 2106ms
    ✓ should display loading message initially
    ✓ should display comments after loading 1039ms
    ✓ should display error message on fetch failure 1016ms

Test Files  1 passed (1)
Tests       3 passed (3)
Start at    23:29:45
Duration    2.28s
```

Explicació:

- Com al codi anterior, cal fer una sèrie d'importacions abans de començar la prova, com per exemple el mount, que prové de **vue/test-utils**
- Els testos s'agrupen en blocs describe.
- **beforeEach**: Aquesta funció s'executa abans de cada prova individual. Aquí, muntem el component Comments i assignem la instància a wrapper.
- Les proves inicials que realitza són les següents:
  - Verificar el missatge de càrrega inicial
  - Verifica que es mostren els comentaris
  - Verifica els missatges d'error

Codi:

```
import { mount } from '@vue/test-utils'
import Comments from '../Comments.vue'

describe('Comments.vue', () => {
  let wrapper

  beforeEach(() => {
    wrapper = mount(Comments)
  })

  it('should display loading message initially', () => {
    expect(wrapper.find('.text-xl').text()).toBe('Carregant comentaris...')
  })

  it('should display comments after loading', async () => {
    // Simular la càrrega de comentaris
    await wrapper.vm.$nextTick() // Esperar que el component es monti
    await new Promise(resolve => setTimeout(resolve, 1000)); // Simular un retard en la càrrega

    // Verificar que els comentaris es mostren
    const comments = wrapper.findAll('.bg-pink-200')
    expect(comments.length).toBeGreaterThan(0) // Comprovar que hi ha comentaris
  })

  it('should display error message on fetch failure', async () => {
    // Simular un error en la càrrega de comentaris
    global.fetch = vi.fn(() =>
      Promise.reject(new Error('Error carregant comentaris'))
    )

    wrapper = mount(Comments) // Tornar a muntar el component per a la nova simulació
    await wrapper.vm.$nextTick() // Esperar que el component es monti
    await new Promise(resolve => setTimeout(resolve, 1000)); // Simular un retard en la càrrega

    // Verificar que es mostra el missatge d'error
    expect(wrapper.find('.text-red-500').text()).toBe('Error carregant comentaris')
  })
})
```

# Users

En aquest últim test, estudiarem el funcionament de d'Users, la qual agafa una llista d'usuaris d'una API.

```
DEV v2.1.4 /home/alumne/Documents/2nDAM/MP07 - Desenvolupament d'interfícies/Projecte 1/ProjecteJavaScript1/project

✓ src/views/test/users.spec.js (2)
  ✓ User List Component (2)
    ✓ fetches and sets users correctly
    ✓ handles errors correctly

Test Files 1 passed (1)
Tests 2 passed (2)
Start at 00:34:32
Duration 699ms (transform 22ms, setup 0ms, collect 98ms, tests 14ms, environment 248ms, prepare 127ms)
```

Explicació del codi:

- Primerament declara el codi i gestiona l'estat en **ref:users, loading i error**
- Seguidament carrega les dades en **unMounted** es fa una petició GET en axios.
- Renderització condicional, finalment al template es mostra un missatge de loading si es true

```
import { ref, onMounted } from 'vue'
import { describe, it, expect, beforeEach, vi } from 'vitest'
import axios from 'axios'
import axiosMockAdapter from 'axios-mock-adapter'

const users = ref([])
const loading = ref(true)
const error = ref(null)

const fetchUsers = async () => {
  loading.value = true
  error.value = null
  try {
    const response = await axios.get('https://dummyjson.com/users')
    users.value = response.data.users
  } catch (e) {
    error.value = 'Error carregant usuaris'
  } finally {
    loading.value = false
  }
}

describe('User List Component', () => {
  let mock

  beforeEach(() => {
    mock = new axiosMockAdapter(axios)
    users.value = []
    loading.value = true
    error.value = null
  })

  it('fetches and sets users correctly', async () => {
    mock.onGet('https://dummyjson.com/users').reply(200, {
      users: [
        { id: 1, firstName: 'John', lastName: 'Doe', email: 'john@example.com' },
        { id: 2, firstName: 'Jane', lastName: 'Doe', email: 'jane@example.com' }
      ]
    })

    await fetchUsers()

    expect(loading.value).toBe(false)
    expect(error.value).toBe(null)
    expect(users.value.length).toBe(2)
    expect(users.value[0].firstName).toBe('John')
    expect(users.value[1].email).toBe('jane@example.com')
  })

  it('handles errors correctly', async () => {
    mock.onGet('https://dummyjson.com/users').reply(500)

    await fetchUsers()

    expect(loading.value).toBe(false)
    expect(error.value).toBe('Error carregant usuaris')
    expect(users.value.length).toBe(0)
  })
})
```